

FPGA+DSP 异构视频处理系统中 基于 SRIO 的数据高效传输方法

姜宏旭 刘亭杉 李辉勇 张 萍 段沐毅

(北京航空航天大学计算机学院数字媒体北京市重点实验室 北京 100191)

摘 要 数据传输一直是影响嵌入式视频系统实时处理能力的关键环节. 随着视频应用的多路化和高清化, 混合多处理器结构已成为嵌入式视频处理系统的主要发展趋势, 异构处理器之间数据的高效传输对系统性能的影响变得比以前更加突出. 文中针对 FPGA+DSP 异构视频处理系统中的数据传输问题, 在分析处理器结构和视频数据格式特征的基础上, 提出了一种基于高速串行接口 SRIO(Serial Rapid I/O)的数据高效传输方法. 该方法分别以 FPGA、DSP 作为系统的传输、处理核心, 在 FPGA 处理器上采用视频三分量数据重组方法并使用包头信息较小的 SRIO 流写事务 SWRITE(Streaming Write), 简化视频传输格式的同时提高了 SRIO 视频数据包的传输效率; 在 DSP 处理器上通过预定义接收端数据存储单元和采用简洁的 SRIO 门铃事务(DOORBELL)应答机制, 节约了 DSP 在传输过程中的时间开销. 实验结果表明, 文中设计的 SRIO 高效传输方法在占用较少的 FPGA 资源的条件下传输速度达理论值的 81% 以上.

关键词 FPGA+DSP; 视频数据; SRIO; 高效传输

中图法分类号 TP399

DOI号 10.11897/SP.J.1016.2015.01119

A High-efficiency Data Transmission Method Based on SRIO in FPGA+DSP Heterogeneous Video Processing System

JIANG Hong-Xu LIU Ting-Shan Li Hui-Yong ZHANG Ping DUAN Mi-Yi

(Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191)

Abstract Data transmission is a critical part of an embedded video system for its real-time processing ability. With the development of digital video processing applications, in order to alleviate the requirement of large amounts of data transmission (take multi-channel, high-definition video for example), multi-processor technique will become more popular in the future. In this paper, we proposed a novel high-efficiency data transmission method based on Serial Rapid I/O (SRIO) for FPGA+DSP video processing system. Considering the heterogeneous processor structure and video data format, we made FPGA as a core unit of transmission and DSP as for processing. Specially, we reorganized the video transmission sequence after separating the three components of video data and applied a lightweight SRIO streaming write transaction (SWRITE) which has minimal header overhead to increase data throughput. To reduce the DSP transmission time, the video data was directly stored in the predefined DSP storage unit and a simple DOORBELL transaction was used as a notification message. The experimental results show that our method can save lots of FPGA resources, and at the same time the transmission rate can achieve 81% of the SRIO theoretical value, which overcomes most of existing methods.

Keywords FPGA+DSP; video data; SRIO; high-efficiency transmission

收稿日期:2014-07-18;最终修改稿收到日期:2015-01-07. 本课题得到国家自然科学基金(61272347)、国家杰出青年科学基金(61125206)资助. 姜宏旭,男,1976年生,博士,副教授,主要研究方向为数字媒体处理及嵌入式系统设计. E-mail: jianghx@buaa.edu.cn. 刘亭杉,女,1986年生,博士研究生,主要研究方向为数字媒体处理及嵌入式系统设计. 李辉勇,男,1983年生,博士研究生,主要研究方向为数字媒体处理及嵌入式系统设计. 张 萍,女,1987年生,硕士研究生,主要研究方向为数字媒体处理及嵌入式系统设计. 段沐毅,男,1953年生,博士,教授,主要研究领域为数字视频/图像压缩处理技术、信息融合技术等.

1 引言

视频应用领域的快速增长和传感器技术的迅速发展推动了视频时空分辨率的提高,同时也使多路化、高清化成为视频应用发展的一种趋势.视频数据量的激增不仅给处理平台的计算能力带来了严峻挑战,同时也给视频处理系统中处理器间数据的高速传输造成了极大困难.例如,当使用 YCbCr 4:2:0 采样格式采集视频数据时,一路分辨率为 720×576 ,帧率为 25 帧/秒的 D1 数字视频原始数据率约为 119 Mbps,一路 1080P30 高清数字视频的原始数据率高达 712 Mbps.因此,在视频处理系统设计中除了要关注处理器的性能和计算能力之外,研究和解决处理器间视频数据的高效传输问题同样重要.

目前,嵌入式视频处理系统中常用的互连和传输接口 EMIF(External Memory Interface)等已难以满足多路或高清视频的大数据量传输需求^[1-3].高速串行接口 SRIO 作为 RapidIO 的一个重要分支,是面向嵌入式系统开发提出的高可靠、高性能、基于包交换的新一代高速互联技术,SRIO 技术的发展为异构处理器间的高速互联提供了另一种可能.特别是近年来推出的数字信号处理器(Digital Signal Processor, DSP)和现场可编程门阵列(Field Programmable Gate Array, FPGA)等嵌入式处理器普遍集成高速差分串行收发器,很大程度上促进了 SRIO 的发展应用^[4-7].尽管 SRIO 具有可选的 1.25 Gbps、2.5 Gbps、3.125 Gbps 等多种传输速率,然而这只是理论值,低效的传输机制设计或是繁琐的实现过程都会导致其传输速度的大幅降低.例如:文献[8]基于多核 DSP 设计实现了声纳基阵处理系统,此系统使用 SRIO 的普通读事务 NREAD(Normal Read)、普通写事务 NWRITE(Normal Write)、门铃事务 DOORBELL 以及对 RapidIO 内部寄存器进行操作的 MAINTENCE 事务,传输方法上未加优化,测试 1.25 Gbps 工作模式下传输速率约为 257 Mbps,仅达到 SRIO 理论值的 25%;文献[9]通过在 FPGA 端采用乒乓存储结构,并使用带响应的写事务 NWRITE_R(Normal Write with Response)与 DOORBELL 等 SRIO 事务设计传输机制,数据率比未加优化的方法有了明显提升,达到 SRIO 理论带宽的 74%.

由于现有方法在设计 SRIO 传输机制时缺乏针对视频数据和异构处理器结构特点的考虑,直接采

用已有技术难以满足异构视频处理系统中视频的高效传输和处理需求.此外,现有视频处理系统大多偏重于功能的实现,硬件资源的开销较大.例如,文献[10]提出的方法需要为 FPGA 外挂两片同步动态存储器 SDRAM(Synchronous Dynamic Random Access Memory),增加资源占用的同时,加大了系统软硬件设计的复杂度.文献[11]采用基于 DSP 主动发送数据的传输方案,该方案需要 FPGA 向 DSP 发送中断信号以请求数据,每中断一次 DSP 可以发送 1/4 帧的视频数据,因此显示一帧视频需要 4 次中断.由于 DSP 在进行视频处理工作的同时还要负责数据传输,无疑加大了 DSP 时间与资源的开销.

本文在分析视频数据和异构处理器结构特点的基础上提出了一种基于 SRIO 总线的数据高效传输方法.该方法通过在 FPGA 处理器上采用视频三分量数据重组方法并使用包头信息较短的 SWRITE 事务,简化视频传输格式的同时提高了 SRIO 上视频数据包的传输效率;在 DSP 处理器上通过预定义接收端数据存储单元和采用简洁的 DOORBELL 应答机制,节约了 DSP 在传输过程中的时间开销.本文方法由于仅采用 SWRITE、NREAD 和 DOORBELL 这 3 种 SRIO 事务,减化了实现过程的复杂度.此外,由于视频数据是以行为单位在 FPGA 上存储的,并且在存储过程中,采用 FIFO 视频缓冲队列取代资源占用率较高的 RAM,因此系统对 FPGA 内存资源占用较少,无需外部存储器,降低了软硬件设计的难度.实验证明,本文使用 SRIO 单通道并采用 1.25 Gbps 工作速度,传输速率达 829 Mbps 以上,可满足六路 D1 视频或一路 1080p 高清视频的传输需求.文中以一路 D1 视频传输为例做出说明及验证方法的可行性.

本文第 2 节对系统的整体概况进行描述,包括本文所采用的视频处理系统的结构、视频数据格式的分析、SRIO 事务类型的介绍以及 SRIO 传输过程的介绍等内容;第 3 节详细介绍本文提出的异构处理器间 SRIO 高效传输方法,主要分为方法处理流程、SRIO 高效传输机制设计与方法的实现 3 个主要部分;第 4 节首先对实验环境进行介绍,然后分别给出方法的 SRIO 传输速率测试与 FPGA 资源使用情况的相关实验结果.

2 系统概述

2.1 FPGA+DSP 异构视频处理系统结构

由于数据计算能力强、功能定制灵活, DSP 和

FPGA 已成为嵌入式视频处理系统中应用最广泛的两种处理器. 其中, DSP 是一种适合进行数字信号处理运算的微处理器^[12], 主要用来实现各种数字信号处理算法; FPGA 虽然具有体系结构灵活、集成度高以及适用范围广等特点^[13], 但相对于 DSP 而言, 其算法程序的设计难度大、开发周期较长. 现有主流的基于多处理器的视频处理系统主要有基于 DSP + DSP 同构处理器架构和 FPGA + DSP 异构处理器架构两种.

相比之下,同构处理器架构在设计上比异构架构下的视频处理系统简单,但是处理器大多只能选择带有视频接口的专用 DSP(例如 TI 的 DM64× 系列和 DaVinci 系列),这类处理器除了具有专用的视频接口外,往往还在内部集成了专用协处理器用于完成标准视频处理算法(如 H. 265 编码算法),但由于此类处理器属于半定制类型,其灵活性较差并且对视频处理算法的限制大,因此 DSP+DSP 同构处理器架构主要面向标准视频处理应用领域。而 FPGA+DSP 异构处理器架构在视频接口的适应性和算法设计的灵活性方面更具有优势:首先,FPGA 作为协处理器可以完成视频接口的扩展、格式转换、预处理和系统的控制工作^[14-17],由于 FPGA 具有完全可定制的特点,因此在接口设计和协处理方面比同构处理器结构具有更高的灵活性;其次,DSP 处理器可以从接口和控制任务中脱离出来,只作为主处理器专注于完成大量视频数据的计算和处理。此外,FPGA+DSP 异构处理器架构下的 DSP 在芯片选择上可以不局限于专用 DSP,因此能够选用更高性能的通用 DSP 处理器,大大提高了系统的处理能力和算法设计的灵活性。

本文设计了 FPGA+DSP 异构视频处理系统。在该系统中,根据 FPGA 与 DSP 处理器的各自特点,并综合考虑视频处理系统的常用任务的特点及处理器间负载均衡的要求,对处理任务进行了划分。其中,FPGA 主要用于视频数据的采集、格式转换、预处理、数据传输和逻辑控制等任务;DSP 主要用于视频的处理、分析和编码等核心工作。在存储方面,FPGA 处理器采用以视频行为单位完成数据采集和预处理的方法,其片上存储资源可以满足需求,避免了增加片外存储器;而 DSP 在计算过程中需要存取大量视频数据和中间处理结果,片内存储远无法满足要求,因此为其设计了专门的 DDR3 片外存储器。图 1 所示的 FPGA+DSP 混合处理器的硬件结构具有广泛的通用性,可以作为多种视频处理系

统的核心硬件单元,例如视频分析、处理和编码系统等。

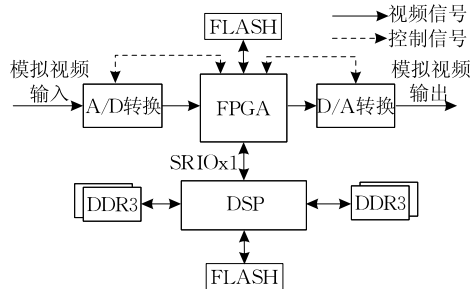


图 1 FPGA+DSP 异构视频处理系统硬件结构

2.2 视频数据格式

视频数据除有一般大数据的规模性和高速性特点外,在结构上还具有独特的规则性和统一性.例如,当前普遍采用的 PAL 制式 ITU-R BT. 656 数字视频在行和帧结构上就具有严格的规则性.图 2 和图 3 分别给出了该类数字视频在 YCbCr 4:2:2 通用采样格式下的行结构和帧结构.

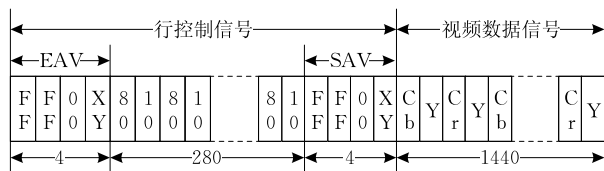


图 2 BT.656 行视频数据结构

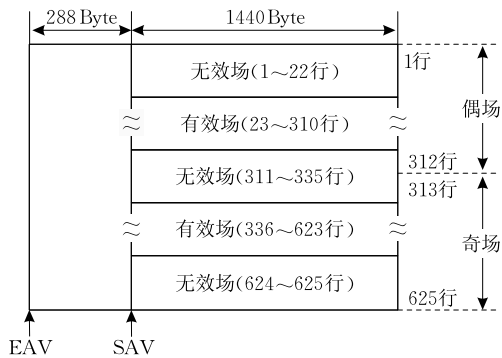


图 3 BT.656 帧数据结构

每个视频行包含了行控制信号和视频数据信号两部分。其中,行控制信号为每个视频行起始的 288 字节,包括前四字节的视频有效信号 EAV (End of Active Video), 280 字节的固定填充数据, 以及最后 4 字节的视频有效帧开始信号 SAV (Start of Active Video); 视频数据信号由 Y、Cb、Cr 3 种视频分量构成, 每行视频含有 720 字节 Y, 360 字节 Cb 和 Cr, 共 1440 字节。其排列顺序为 Cb, Y, Cr, Y, Cb, Y, Cr, ..., 其中, 前 3 个 Cb, Y, Cr 采样的是一个数据点的亮度信号和两个色差信号, 而接下来

的 Y 采样是下一个点的亮度信号,以此类推。

完整的一帧 BT. 656 视频数据是由源视频经过行扫描形成的奇场和偶场组成,如图 3 所示。PAL 制式中第 1~312 行表示偶场,第 313~625 行表示奇场,其中有效场为第 23~310 行和第 336~623 行,可使用 SAV 和 EAV 中的标志信号来表示有效场和无效场。

视频数据除了在采集端具有严格的规则性外,在处理过程中对数据也有统一性的要求。例如, YCbCr 4:2:0 是目前视频处理系统中采用最为广泛的视频格式^[18], 3 种视频分量一般是按帧被依次独立处理。因此,在本文视频采集过程中, FPGA 处理器首先要将按点依次采集的 YCbCr 4:2:2 视频数据转换为按行和帧排列的 YCbCr 4:2:0 格式。在 YCbCr 4:2:2 格式中,隔点分别采集 Cb 和 Cr 分量,在 YCbCr 4:2:0 格式中奇数行隔点采样 Cb 分量,偶数行隔点采样 Cr 分量,如图 4 所示,图中每个方格代表一个像素点,每个圆形代表一种分量,两种格式中采集所有点的 Y 分量。

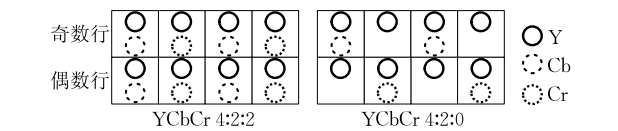


图 4 隔行扫描视频两种 YCbCr 采样格式

针对视频数据在结构上的规范性特点以及处理上的统一性要求,本文为 FPGA 与 DSP 处理器上的收发缓冲区各自分配 3 个视频缓冲队列,以实现视频数据三分量的快速分离,并为两处理器上帧数据的高效整合提供了条件。首先,在分离过程中同时完成视频格式转换工作(YCbCr 4:2:2 转为 YCbCr 4:2:0),即采集到的视频数据按照奇场和偶场及视频分量顺序依次输入,舍弃奇场中 Cr 数据和偶场中 Cb 数据,剩余视频数据按照分量不同存入对应视频缓冲队列。其次,在预定义好 DSP 接收端数据存储地址后,对分离后的视频三分量进行打包传输,使得每种视频分量存到 DSP 指定地址中形成整帧数据结构,从而减少 DSP 计算数据存储地址上的时间开销。

2.3 SRIO 事务操作

为了适应多种类型数据的高速传输,SRIO 从 IO 逻辑(Input Output Logical)与消息传递(Message Passing)两个层面提供了如表 1 所示的 8 种事务功能,这些事务涵盖了从原子到数据包的广泛应用方式。其中,IO 逻辑包括 Atomic、NREAD、NWRITE、

NWRITE_R、SWRITE 与 MAINTENANCE 几种;消息传递包括 DOORBELL 与 MESSAGE 两种。

表 1 SRIO 事务列表

SRIO 事务名称	事务类型	描述说明
Atomic, NREAD	IO 逻辑	普通读操作
Atomic, NWRITE, NWRITE_R	IO 逻辑	普通写操作,带响应的写操作
SWRITE	IO 逻辑	流写操作
MAINTENANCE	IO 逻辑	通过维护端口访问寄存器
DOORBELL	消息传递	发送中断或信号量
MESSAGE	消息传递	分割数据包并一次发送多个

Atomic 为原子操作;NREAD 事务为普通读操作,可以直接从对端内存取数,一包最大为 256 Bytes;普通写操作有 NWRITE 和 NWRITE_R 两种,可以直接往对端内存写数,一包最大为 256 Bytes;SWRITE 事务具有包头信息短、传输数据是 8 Bytes 的整数倍、长度不能超过 256 Bytes 等特点;MAINTENANCE 用于对 RapidIO 内部寄存器的读写操作,可以配置整个 RapidIO 网络上各个节点的参数;DOORBELL 的包头和携带的信息都很短,用于提供一个快速的短消息通知;MESSAGE 用于传输大数据包,每次最多 4 KBytes 数据。

在分析 SRIO 事务特点基础上,结合视频处理系统中视频数据的高速传输需求,本文对 SRIO 的传输机制进行了定制和优化。在 SRIO 事务选择上,由于系统中需要由 FPGA 读取 DSP 处理后的视频数据,因此选择了常规的 NREAD 事务。3 种写事务中,尽管 NWRITE_R 可有效防止数据包的丢失,提高数据包写入的正确率,但是由于本文是通过地址偏移的方式向 DSP 写入视频数据的,该方法不会因为数据包的丢失而造成视频数据的累积错误,因此无需选用设计较为复杂的 NWRITE_R 事务。相比 NWRITE 事务,SWRITE 写事务具有包头信息短、打包率高及实现简单等特点,因此本文 SRIO 写事务选择 SWRITE 事务。由于本系统中 SRIO 仅用于两个异构处理器间点到点的数据传输,并未使用 SRIO 桥芯片,因此无需通过 MAINTENANCE 事务修改和维护 SRIO 收发地址等内部寄存器。此外,为简化 SRIO 传输过程实现的复杂度,选用简洁的 DOORBELL 事务用于异构处理器间的传输控制。综上,本文 SRIO 传输过程中采用 SWRITE、DOORBELL 和 NREAD 这 3 种事务,显著降低了设计实现的复杂度。

2.4 SRIO 传输过程

SRIO 数据传输过程是基于请求和响应机制

的. 发起端请求 IREQ(Initiator Request)是一次事务的开始,用户控制 IREQ 端口组织请求数据包,数据包经过 SRIO 链路发送到目标请求端口 TREQ(Target Request)进行解析,目标端用户控制目标响应端口 TRESP(Target Response)组织响应数据包,数据包经过 SRIO 链路发送回发起端响应端口 IRESP(Initiator Response)解析.

图 5 为一次典型的基于 IP 核的端到端 SRIO 传输过程示意图,图中用户 A 发送 SRIO 请求给用户 B(步骤(1)~(3)),用户 B 响应用户 A 的 SRIO 请求(步骤(4)~(6)),具体步骤如下:

(1)用户程序 A 控制 IREQ 端口信号,填充请求数据包信息,如包类型等;

(2)IP 核 A 根据用户填入信息封装为 SRIO 数据包并发送到 SRIO 链路;

(3)IP 核 B 收到请求数据包,解析后通过 TREQ 端口通知用户程序 B;

(4)用户程序 B 根据请求,控制 TRESP 端口组织响应数据包,填入响应数据;

(5)IP 核 B 封装响应数据为 SRIO 响应数据包并发送到 SRIO 链路;

(6)IP 核 A 收到响应数据包,解析后通过 IRESP 端口通知用户程序 A,用户程序 A 接收并处理响应数据包.

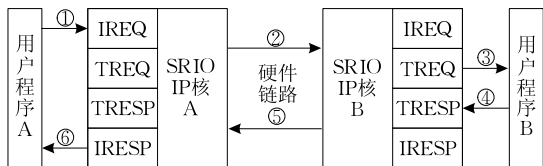


图 5 SRIO 典型传输过程示意

3 异构处理器间 SRIO 高效传输方法

3.1 处理流程

视频编码器多以 Y、Cb、Cr 3 种视频分量的数据块为单位进行编码. 为提高编码效率,避免反复搬运数据造成 DSP 处理时间和资源的浪费,本文在 DSP 接收数据时,为其设计了 3 个接收缓冲区用于视频数据三分量的分离,在 FPGA 端设计 3 个发送缓冲队列与之对应. 对于采集的视频数据, FPGA 首先根据其原始数据的 Y、Cb、Cr 三通道格式组织并按视频序列流打包生成待发送的数据包,然后采用包头信息较短的 SWRITE 事务实现 FPGA 向 DSP 写数据的操作,不同通道的视频序列直接写入 DSP

端预定义的 DDR3 缓冲区,当 SWRITE 事务操作完成后, FPGA 发送 DOORBELL 事务通知 DSP 解析视频数据并进行处理计算, FPGA 向 DSP 发送数据过程如图 6 所示.

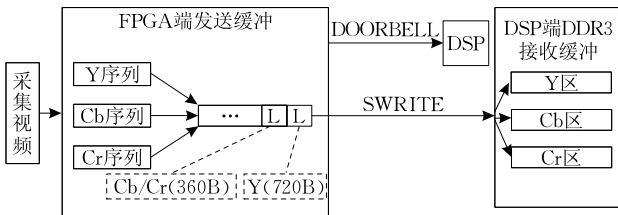


图 6 FPGA 向 DSP 发送数据示意

当 DSP 处理完一帧视频数据后,通过 EDMA (Enhanced Direct Memory Access) 将处理后的视频数据搬运到 DSP 端 DDR3 的输出缓冲区,在 FPGA 端设计 3 个接收缓冲队列与之对应. 当 DSP 处理完接收到的视频数据后,通过向 FPGA 发送 DOORBELL 事务通知 FPGA 收取处理好的视频数据,最后由 FPGA 通过 NREAD 事务请求读回处理后的视频数据, FPGA 从 DSP 接收数据过程如图 7 所示.

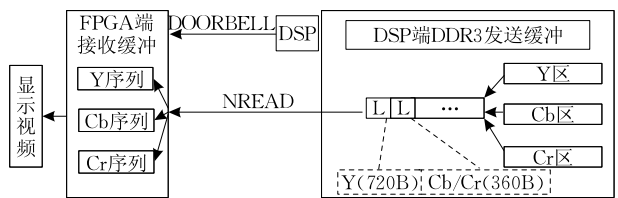


图 7 FPGA 从 DSP 接收数据示意

方法以 FPGA 作为传输链路核心, DSP 作为视频数据处理核心,分别设计 FPGA 和 DSP 端 SRIO 程序,其中, FPGA 端程序包括发起端请求 IREQ 和发起端响应 IRESP; DSP 端程序包括目标端请求 TREQ 和目标端响应 TRESP. 如图 8 所示, SRIO 传输过程中仅采用 SWRITE、DOORBELL 和 NREAD 这 3 种事务,显著降低了设计实现的复杂度. 下面就 SRIO 传输机制和方法实现两方面加以详细介绍.

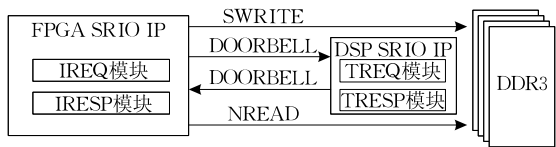


图 8 异构处理器间 SRIO 事务传输

3.2 SRIO 高效传输机制设计

为了实现 SRIO 的高效传输,本文在 FPGA 端首先对采集到的视频数据进行发送前的重新组织,

如图 9 所示. 具体组织过程如下, FPGA 采集打包格式的 YCbCr4:2:2 视频数据, 为 3 种分量设置各自视频缓冲队列, 用于分离并暂存视频分量. 在分离过程的同时完成视频格式转换工作 (YCbCr 4:2:2 转为 YCbCr4:2:0). 采集到的视频数据按照奇场和偶场及视频分量顺序依次输入, 舍弃奇场中 Cr 数据和偶场中 Cb 数据, 剩余视频数据按照分量不同存入对应视频缓冲队列. 当一个完整视频行的分量数据都写入各自视频缓冲队列后, 分别读取各视频缓冲队列, 以视频行为单位将 720 字节 Y 分量和 360 字节 Cb (或 Cr) 分量写入 SRIO 发送缓冲队列 (TxFifo). 至此, SRIO 发送缓冲队列 TxFifo 中已经存入以平面格式视频行 L 为单位的 YCbCr 4:2:0 视频数据, 等待发送.

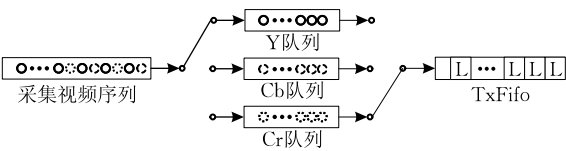


图 9 视频数据组织示意

由于 SRIO 链路是以包为单位传输数据的, 因此为了提高数据包的传输效率, 在数据发送过程中采用包头信息较短的 SWRITE 事务, 设置其有效字节为 SRIO 协议所规定的最大字节数 256 Bytes, 此时该事务有效数据打包率为 95%, 能达到最快的 SRIO 写速率. 在传输过程中, 需要为每个 SWRITE 事务包设置特定地址和数据长度, 视频数据以行为单位进行循环发送. 由于 SRIO 数据包最大有效数据载荷为 256 Bytes, 在保证数据完整性并满足 SWRITE 传输数据是 8 Bytes 整数倍要求的前提下, 为最大程度提高 SRIO 传输效率, 应尽量减少数据包个数. 因此, 本文将 720 Bytes 的 Y 数据划分为 3 个 SWRITE 包, 每包传输 240 Bytes, 包存储地址可以由帧基地址、Y 分量偏移量和行偏移量计算获得, 3 个包存储地址连续增加; 360 字节 Cb (或 Cr) 数据采用两个 SWRITE 事务包发送, 并规定两个数据包的有效数据长度分别为 184 Bytes 和 176 Bytes, 存储地址由帧基地址、Cb/Cr 分量偏移量和行偏移量计算获得.

为了提高 DSP 的视频处理效率, 在本异构视频处理系统中, 采用由 FPGA 发送 NREAD 读请求方式读取 DSP 处理后的视频数据, DSP 仅需对 NREAD 请求进行读响应即可, 该响应过程对 DSP 是透明的, 如此可极大程度减小 DSP 在传输过程中的时间与资源开销, 使得 DSP 可以专注于视频处理工作. 其传输过程与 SWRITE 事务类似, 即为每个

NREAD 事务包设置特定地址和长度, 视频数据按行循环发送. 为最大程度提高 SRIO 写事务效率, 参照 SWRITE 事务的数据包划分方式以及包存储地址的计算方式, 将 720 Bytes 的 Y 数据划分为 3 个 NREAD 包, 每包传输 240 Bytes, 包存储地址可以由帧基地址、Y 分量偏移量和行偏移量计算获得, 3 个包存储地址连续增加; 360 字节 Cb (或 Cr) 数据采用两个 NREAD 事务包发送, 并规定两个数据包的有效数据长度分别为 184 Bytes 和 176 Bytes, 存储地址由帧基地址、Cb/Cr 分量偏移量和行偏移量计算获得.

此外, 预定义 DSP 端数据存储单元是传输机制设计的另一重要组成, 如图 10 所示. 预定义 DSP 端数据存储单元包括 SWRITE 和 NREAD 事务的存储地址计算. 由于此两种事务地址计算方法相同, 故下面以 SWRITE 事务为例进行说明. FPGA 发送的每个 SWRITE 事务包中携带的数据将被按顺序写入 DSP 的视频缓冲区中的连续存储空间, 为实现高效 SRIO 传输过程应使用高载荷 SWRITE 事务包. 因此, 在综合考虑采集视频过程特点和视频存储计算特点后, 规定:

- (1) TxFifo 中数据按视频行 L (Line) 排列, 每行按照 720 字节 Y、360 字节 Cb (或 Cr) 数据顺序排列;
- (2) 每个 SWRITE 事务包中数据只能包含一种分量 (即 Y/Cb/Cr 中的一种), 数据存储地址依次连续递增;
- (3) 在 DSP 视频缓冲区中数据按视频帧 F (Frame) 存储, 每帧数据中同一分量连续存储, 组成各分量矩阵, 便于 DSP 完成视频处理.

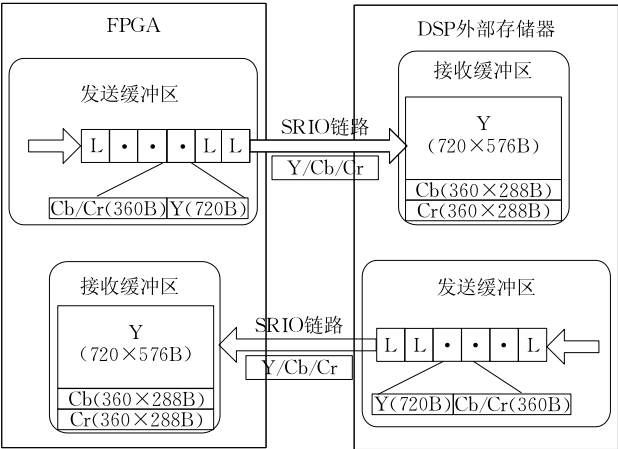


图 10 SRIO 数据存储单元示意

3.3 低复杂度 SRIO 传输方法实现

本文在传输视频数据过程中仅使用 SWRITE、

NREAD 和 DOORBELL 这 3 种 SRIO 事务,其中 DOORBELL 事务仅用来发送中断信号,操作相对 MESSAGE 更为简洁.同时,使用逻辑与控制较 RAM(Random Access Memory)简单的 FIFO(First In First Out)存储机制也在一定程度上降低了设计的复杂度,并且节约了 FPGA 的资源占用.具体实现过程如下:

(1) 在视频数据由 FPGA 写入 DSP 过程中, FPGA 作为主动发起端,将模数转换后的数字视频通过 SWRITE 事务写入 DSP 对应的 DDR3 存储区,为保证内存访问的正确性,视频在写 DDR3 存储区时, DSP 不得对该存储区进行操作.当 FPGA 写完一帧数据之后,发送一个 DOORBELL 事务,通知 DSP 可以对该帧数据进行处理,保证 FPGA 与 DSP 之间的同步.为了保证 DSP 上处理的连续性,在 DSP 上设置了 8 帧缓冲区,可以循环对数据帧进行处理.

(2) 在视频数据由 DSP 写入 FPGA 的过程中,每当 DSP 处理完一帧视频之后,发送 DOORBELL 事务通知 FPGA 可以进行数据读取. FPGA 接收到该同步信号之后,作为主动发起端,将 DSP 处理后的视频数据通过 NREAD 事务读入 FPGA 的 RxFifo 缓存中.在实际运行过程中,为保证数模转换后视频的连续性,必须维持 FPGA 端 RxFifo 缓存不空,同时要兼顾数据采集的帧率.方法中所采集的模拟视频帧率为 25 帧/秒,即每 40 ms 采集一帧.但在实际采集过程中,帧率会受到摄像头采集图像的复杂程度影响而产生偏差,为了防止出现同步问题,规定在 DSP 处理完两帧数据后向 FPGA 发送 DOORBELL,使其进行读操作.在读的过程中, FPGA 只需维护自己的 RxFifo 即可.在设计状态机实现读写功能时,以读优先的方式进行操作,设置 RxFifo 的 Prog_empty 信号为 2000 Bytes,即当 RxFifo 中的数据小于 2000 Bytes 时,进行读操作,否则,进行写操作.在读数据的过程中,不能进行写操作,但此时模拟视频却仍然在实时输入,为了保证读数据的时间内采集用的 Tx FIFO 缓冲区不溢出需要计算 Tx FIFO 的最大值.

具体控制过程如下:

(1) SWRITE 事务的触发:由于视频是以行为单位进行采集并传输到 Tx FIFO 的,因此,为保证行数据结构的完整性,当 Tx FIFO 中数据不少于一行(720 字节 Y, 360 字节 Cb 或 Cr)且此时没有读请求时,可发送数据,即通过 SWRITE 事务完成一次写

操作.

(2) DOORBELL 事务的触发:在 FPGA 向 DSP 发送数据时,当完成一帧数据的传输且此时没有读请求时,即可通过发送 DOORBELL 事务通知 DSP 处理接收到的当前帧数据;在 FPGA 从 DSP 读取数据过程中,当 DSP 处理完两帧视频数据时发送 DOORBELL 事务通知 FPGA 读取第一帧数据.

(3) NREAD 事务的触发:以 FPGA 端接收缓冲区 Rx FIFO 中数据量和 DSP 发送给 FPGA 的 DOORBELL 事务为触发条件,当 Rx FIFO 中有多于一行的剩余空间即可组织 NREAD 事务完成一行数据的读取.

另外,规定 3 种事务发送优先级由高到低分别为 NREAD、DOORBELL 和 SWRITE 事务.当 NREAD 事务发出后, DSP 的 SRIO 硬件接口开始组织响应数据包,响应数据包数据量大,时间紧迫程度高,需要尽早开始响应数据包组织,同时 NREAD 和 DOORBELL 事务占用发送周期少,因此 NREAD 发送优先级最高; DSP 收到 DOORBELL 事务后解锁视频帧存储区,开始视频处理工作,因此 DOORBELL 优先级高于 SWRITE 事务.

Xilinx 公司的 SRIO IP 核提供 IREQ、IRES、TREQ、TRESP 4 个端口.由于本文方法 FPGA 在读写过程中都是作为传输数据包的发起端,而 DSP 作为响应端只需接收和发送简洁的 DOORBELL 信号,其 TREQ 与 TRESP 端口实现相对简单.因此,传输过程设计的难度主要集中在 FPGA 端的 IREQ 与 IRES 两个端口.其中, IREQ 模块做为一次事务的开始,负责组织请求数据包; IRES 模块负责接收 NREAD 的响应数据包.下面将对此两个模块的设计进行详细介绍.

3.3.1 发起端请求(IREQ)设计

发起端请求模块包含以下 4 部分: CON(Controller)、TIK(Tickler)、IGEN(IREQ Generator)和 Tx FIFO.其中, CON 是整个 IREQ 模块的核心控制部分,是一次 SRIO 请求事务包的起点,它负责检测是否发送数据、控制何时发送数据以及组织数据包信息, CON 监控 Tx FIFO 状态,控制 TIK 完成数据包发送; TIK 负责具体的数据组织工作,产生数据包包头信息,并将包头信息传递到 IGEN, 监控 IGEN 状态并向 CON 反馈; IGEN 负责与 IP 核的 IREQ 端口协同将组织好的数据包信息传递给 IP 核,完整的数据包包括包头信息和数据信息, IGEN 的包头信息来自 TIK, 数据信息可以直接取自 Tx FIFO 或接

收 TIK 传递的数据;TxFifo 是发送数据缓冲队列. SRIO 用户程序 IREQ 模块结构如图 11 所示,视频数据依次流经 TxFifo 和 IGEN,最终由 IGEN 传递到 IP 核的 IREQ 端口并通过链路发出.

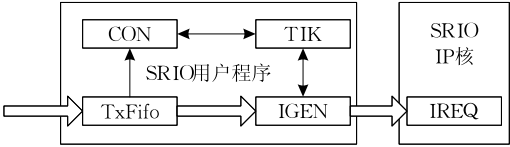


图 11 SRIO 用户程序 IREQ 模块结构示意图

CON 是 FPGA 端所有 SRIO 请求的发起者,也是 FPGA 控制 SRIO 通路数据收发的核心,使用 CON 模块请求事务状态机完成 SRIO 通路控制工作,如图 12 所示. 状态机包含 6 种状态:INIT (Initial)、RDY (Ready)、SW (SWRITE)、DB (DOORBELL)、NR (NREAD)和 SEND 状态.

(1) INIT 状态是状态机的初始状态,当 SRIO 硬件初始化成功后进入 RDY 状态,否则继续等待;

(2) RDY 状态表示 SRIO 链路正常,当发送请求来临并且 IP 核就绪时,进入对应发送请求状态,若 SRIO 链路异常,则进入 INIT 状态;

(3) NR、DB 和 SW 状态分别是发送 NREAD 事务、DOORBELL 事务和 SWRITE 事务状态,每种事务分别组织各自的有效事务信息,当 IP 核状态就绪时进入 SEND 状态,否则等待,若 SRIO 链路出现异常或者中断,则进入 RDY 状态;

(4) SEND 状态为 SRIO 请求事务发送状态,CON 模块将组织好的数据信号传递给 TIK 模块,命令 TIK 模块开始组织数据包的包头信息,并发送 SRIO 数据包,当 TIK 完成数据接收则进入 RDY 状态,否则继续等待,若出现链路异常或者中断,则进入 RDY 状态.

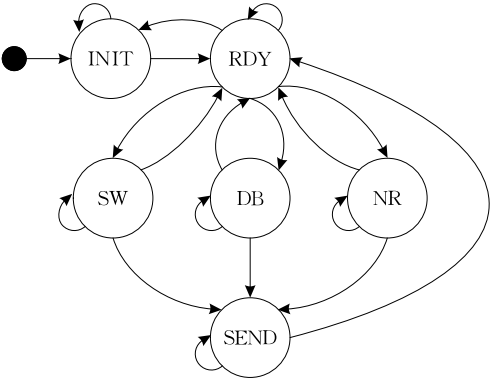


图 12 CON 模块请求事务控制状态机示意图

3. 3. 2 发起端响应 (IRESP) 设计

SRIO 用户程序的 IRESP 模块主要负责接收

并解析 DSP 发送的 NREAD 响应事务包数据,主要包括 IHAN (IRESP Handler) 和 RxFifo 两部分, IHAN 负责接收及解析数据包, RxFifo 是接收数据缓冲队列. IHAN 接收解析数据工作由 IHAN 模块事务解析状态机控制完成,如图 13 所示. IHAN 状态机只包括 3 个状态:INIT、RDY 和 RECV 状态.

(1) INIT 状态是初始状态,表示当前 SRIO 硬件链路连接仍未建立成功,当 SRIO 链路建立成功后进入 RDY 状态,否则等待;

(2) RDY 状态表示当前 IHAN 模块处于空闲状态,可以接收响应事务,当响应事务到来时,进入 RECV 状态,当 SRIO 链路断开时,进入 INIT 状态;

(3) RECV 状态表示当前正在接收 IP 核 IRESP 端口数据信息,同时完成向 RxFifo 写入操作,当数据接收完成时,进入 RDY 状态,当 SRIO 链路端口时,进入 INIT 状态,否则继续接收数据,处于 RECV 状态.

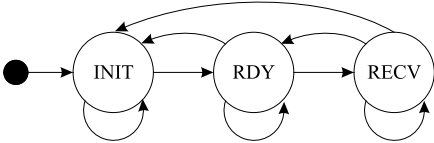


图 13 IHAN 模块事务解析状态机示意

4 实验结果

4. 1 实验环境

本文视频处理系统选用 TI 公司的 TMS320C6678 DSP 芯片及 XILINX 公司的 XC5VSX50T FPGA 芯片. 其中,TMS320C6678 有 8 个 TMS320C66x DSP 核,单核工作频率为 1. 0GHz 或 1. 25GHz,其 SRIO 接口可配置成单通道、双通道或四通道模式,单通道可选择工作在 1. 25 Gbps、2. 5 Gbps、3. 125 Gbps 和 5Gbps 4 种速率模式下^①;XC5VSX50T 芯片包含 12 个 GTP 串行高速收发器、288 个 DSP48E 硬件模块、8160 个 Slices、4752Kb BlockRAM 和 6 个 CMT 时钟管理模块^②.

此外,系统选用 Xilinx 公司提供的 LogiCORE IP Serial RapidIO v5. 6 版本的 SRIO IP 核,此款 IP 核支持 Virtex4、Virtex5 和 Spartan6 等系列 FPGA,共包括以下 3 个功能模块:逻辑与传输层模块、缓冲

① Texas Instruments Incorporated. TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor. <http://www.ti.com>, 2010

② Xilinx Incorporated. Virtex-5 FPGA User Guide. <http://www.xilinx.com>, 2012

区模块及物理层模块. 为方便用户进行操作, 以上 3 个模块被集成到 RIO_WRAPPER 模块中. 此外, RIO_WRAPPER 模块还提供了 RocketIO 封装、时钟、复位和寄存器管理模块, 为系统提供了硬件支持. 除硬件选型外, 系统硬件布线参数也会影响 SRIFO 传输速率, 本文使用文献[19]所提方法对系统进行硬件仿真, 并参考仿真结果完成 PCB 设计, 本 FPGA+DSP 异构视频处理系统实物图如图 14 所示.

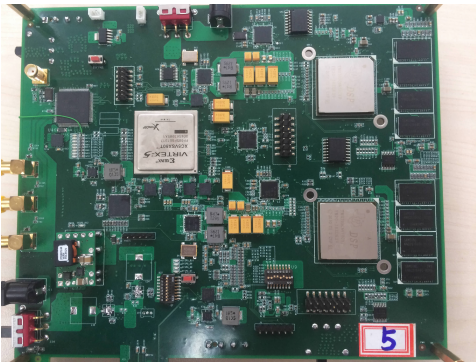


图 14 FPGA+DSP 异构视频处理系统实物

4.2 SRIO 传输速率测试

本文将 SRIO 总线速率设置为 1.25 Gbps 模式, 由于 SRIO 总线物理层采用 8b/10b 编码, 因此 SRIO 传输理论带宽为 1 Gbps. 由于本文提出的方法中 SWRITE 事务、NREAD 事务都是由 FPGA 发起的, 因此, 在 FPGA 端需要按照各种事务优先级顺序分时复用 IREQ 端口进行数据收发; 另一方面, 由于传输过程中的数据打包需要占用一定时间和空间开销, 因此, SRIO 实际传输速率跟理论值是有一定差距的.

基于上述实验设定, 针对 FPGA+DSP 异构视频处理系统, 本文需要测试 FPGA 通过 SWRITE 事务与 NREAD 事务读写 DSP 端 DDR3 存储器的速率, 并对 FPGA 资源使用情况进行测试. 由于受到硬件系统中调试工具的限制, 不能通过计算机实时反馈硬件工作状态, 故在测试本文方法 SRIO 链路发送与接收数据速率时, 通过检查 1 s 时间内 DDR3 存储器中写入数据长度得到 FPGA 向 DSP 发送数据的速率; 类似地, 通过检查 1 s 时间内 FPGA 收到数据包的数量得到 FPGA 读取 DSP 的速率.

(1) SWRITE 事务速率测试

为测试本文方法 FPGA 的 SRIO 发送速率, 在一秒钟时间内 FPGA 不间断发送特定图像数据到 DSP 的 DDR3 存储器中, 发送结束后通过 CCS 工具及 DSP 仿真器导出 DDR3 存储内容, 在计算机上检

查写入数据长度, 该数据长度即是 FPGA 的 SRIO 发送速率.

本文对 SWRITE 事务进行测试, 检查 DDR3 存储内容, 发现写入数据符合递增规律, 数据正确无误, 平均每秒写入数据长度 880 Mb, 即 FPGA 向 DSP 通过 SRIO 发送速率为 880 Mbps, 达到传输速率理论值的 86%, SRIO 写速率测试表如表 2 所示.

表 2 SRIO 写速率测试表

次数	数据大小/Mb	传输时间/s	传输速度/Mbps
1	873	1	873
2	885	1	885
3	881	1	881

(2) NREAD 事务速率测试

为测试本文方法 FPGA 通过 SRIO 读取 DDR3 的速率和正确性, 在一秒时间内, 由 FPGA 向 DSP 连续发送 NREAD 请求事务, 在 FPGA 端接收 NREAD 响应事务, 检查收到数据正确性并记录收到的数据包数量, 通过 ChipScope 工具及 FPGA 仿真器检查数据正确性标志位及数据包数量. 本文测试平均读速度为 829 Mbps, 达到传输速率理论值的 81%, 数据正确无误, SRIO 读速率测试表如表 3 所示.

表 3 SRIO 读速率测试表

次数	数据大小/Mb	传输时间/s	传输速度/Mbps
1	833	1	833
2	815	1	815
3	838	1	838

(3) 方法比较

本节将对 SRIO 的读速率、写速率以及 SRIO 事务类型 3 方面对本文方法与其他文献方法进行对比分析.

表 4 为本文方法与其他文献方法在 SRIO 读写速率方面的对比表. 其中, 文献[8]为未加任何优化的传统方法, 采用 SRIO 的 1.25 Gbps 的工作模式, 由于此文未对传输方法进行优化, 其读写速率均较低, 只能达到理论带宽的 25% 左右. 文献[9]通过在 FPGA 端采用乒乓存储结构保障视频数据传输流畅性的同时, 在 DSP 端预定义了数据接收的存储单元, 此文 SRIO 工作在 3.125 Gbps 的工作模式下, 传输速率为 2320 Mbps, 约占理论带宽的 74%. 从表 2 中可以看出当 SRIO 同样工作在 1.25 Gbps 模式时, 本文方法在读写速率上明显优于文献[8]方法, 可以达到 829 Mbps 以上. 由于文献[9]方法工作在 3.125 Gbps 与本文不同, 故我们针对 SRIO 的读写传输效率进行对比, 本文也具有较明显优势.

表 4 SRIO 传输速率的对比表

	SRIO 写速率/Mbps	SRIO 写效率/%	SRIO 读速率/Mbps	SRIO 读效率/%	SRIO 事务类型
文献[8]方法(1.25 Gbps)	258	25	257	25	NREAD, NWRITE, DOORBELL, MAINTENCE
文献[9]方法(3.125 Gbps)	2320	74	—	—	NWRITE_R, DOORBELL
本文方法(1.25 Gbps)	880	86	829	81	NREAD, SWRITE, DOORBELL

在 SRIO 使用的事务类型方面,文献[8]采用 NREAD、NWRITE、DOORBELL 与 MAINTENCE 4 种 SRIO 事务,比本文多了一种事务,在设计上更为复杂,此外,NWRITE 事务在 IREQ 端需要控制 18 种信号,而本文采用的 SWRITE 事务相比 NWRITE 减少了数据包事务类型、数据包编号、打包字节使能和打包字节计数 4 种信号,因此只需控制 14 种信号即可,也一定程度上降低了设计复杂度.由于文献[9]系统中只涉及 FPGA 向 DSP 写数据,故 SRIO 未使用读相关的事务,但是其使用的 NWRITE_R 为带响应的写操作,实现相对复杂.

4.3 FPGA 资源占用测试

本文 FPGA 作为视频传输的核心控制器,其资源占用情况也是衡量视频数据高效传输的重要指标之一.除 FPGA 的 Slice Registers、Look Up Table (LUT)和 Block RAM/FIFO 这 3 种资源外,数据传输过程中用于 FPGA 与 DSP 间数据读写同步的 DSP 中断次数也会影响传输效率.过多的中断次数会增加 FPGA 与 DSP 间的交互时间,带来 DSP 的编码延迟,从而降低数据传输和处理效率,表 5 为本文方法与其他方法在 FPGA 资源占用上的对比表,考虑到中断次数对传输效率的影响,故在表中加入一行中断次数的统计.

表 5 FPGA 资源对比表

	直接方法		文献[11]方法		本文方法	
	数量	占用率/%	数量	占用率/%	数量	占用率/%
Number of Slice Registers	9217	28	9034	27	8306	25
Number of Slice LUTs	8892	27	8890	27	8372	25
Number of Block RAM/FIFO	91	68	外接 RAM		10	7
DSP 中断次数/帧	>5		5		2	

由表 5 可知,与直接方法和文献[11]方法相比,本文提出的视频传输方法对 FPGA 中 Slice Registers 和 LUT 占用较少.在存储资源方面,由于直接方法是在 FPGA 内部完成视频采集时奇偶场的合并,故需要占用大量的 FPGA 片上存储资源 Block RAM/

FIFO,而本文方法在 FPGA 处理器上只需以行为单位缓存及传输视频数据,视频的奇偶场合并在 DSP 处理器上实现,因此本文方法相比直接方法在 FPGA 存储资源占用上有了明显地减少;相比文献[11]方法,本文方法无需外接 RAM,减少存储资源占用的同时降低了硬件设计的复杂度.在与 DSP 的交互方面,文献[11]方法通过 DSP 中断方式同步 FPGA 与 DSP 之间的数据传输过程,在 FPGA 向 DSP 发送 1 帧视频数据完成后,发送第 1 次中断通知 DSP 进行数据处理,当 DSP 处理完 1/4 帧数据时,由 FPGA 发送第 2 次中断通知 DSP 发送处理后的 1/4 帧数据,以此类推,当 DSP 将完整的 1 帧处理后的数据发送至 FPGA 需要 4 次中断,故此方法 FPGA 发送与接收 1 帧数据共需要 5 次中断;而直接方法中 FPGA 是在检测到 DSP 存储器中有数据后就发中断进行取数,因此读取 1 帧视频数据需要比文献[11]多的中断次数;相比之下,本文提出的方法传输一帧图像仅需要两次中断即可实现 FPGA 与 DSP 的同步.数据的传输过程相对 DSP 完全透明,使 FPGA 能专注于传输控制,DSP 专注于视频处理,有效保障了视频传输效率.

5 总 结

针对 FPGA+DSP 异构处理器结构特征和视频数据格式特征,本文提出了一种以 FPGA 作为传输核心的 SRIO 高效传输方法.该方法通过在 FPGA 处理器上采用视频三分量数据重组方法并使用包头信息较短的 SWRITE 事务简化视频传输格式的同时提高 SRIO 视频数据包的传输效率;通过在 DSP 处理器上预定义接收端数据存储单元和采用简洁的 DOORBELL 应答机制节约了 DSP 在传输过程中的时间开销.在方法实现上,本文仅采用 3 种 SRIO 事务,降低了设计的复杂度.实验结果表明,本文方法在占用较少 FPGA 片上资源的条件下,读写速率可达 SRIO 理论值的 81%以上,可满足六路 D1 视频或单路高清视频传输需求.在实际工程中,可通过提高 SRIO 单通道工作频率或采用多通

道传输等方式进一步提高方法传输速率,以满足更多路标清乃至多路高清视频大数据量的传输需求。

参 考 文 献

- [1] Kang Yan, Wang Heng. Design of a system based on DSP and FPGA for video recording and Replaying//Proceedings of the International Symposium on Photo-electronic Detection and Imaging 2013, Optical Storage and Display Technology. Beijing, China, 2013: 89130E-1-89130E-7
- [2] Zamanlooy B, Vaghef V H, Mirzakuchaki S, et al. A real time infrared imaging system based on DSP&FPGA//Proceedings of the 2nd Pacific Rim Symposium (PSIVT). Santiago, Chile, 2007: 16-23
- [3] Wang Yuan-Peng, Jiang Hong-Xu, Yu Hui-Rong. A multi-mode high-speed video data capture system based on DSP+FPGA//Proceedings of the 2011 International Conference on Multimedia Technology (ICMT). Hangzhou, China, 2011: 5230-5233
- [4] Zhang Feng, Ren Guo-Qiang, Wu Qin-Zhang. A high-speed serial transport platform based on SRIO for high-resolution image. Opto-Electronic Engineering, 2010, 37(10): 89-92 (in Chinese)
(张峰, 任国强, 吴钦章. 基于 SRIO 的高速图像串行传输系统设计. 光电工程, 2010, 37(10): 89-92)
- [5] Zhang Yong, Wang Yong, Zhang Ping. Research on physical layer traffic management schemes in serial RapidIO interconnect. The Journal of China Universities of Posts and Telecommunications, 2011, 18(1): 64-69
- [6] Zhang Feng, Wu Qin-Zhang, Ren Guo-Qiang. A high-speed serial transport platform based on SRIO for high-resolution image//Proceedings of the 3rd International Congress on Image and Signal Processing (CISP). Yantai, China, 2010: 2441-2444
- [7] Huang Xiao-Yun, Su Hai-Bing, Wu Qin-Zhang, Wu Wei. Multi-Processor parallel system based on high-speed serial transceiver//Proceedings of the 2nd International Workshop on Education Technology and Computer Science (ETCS). Wuhan, China, 2010: 178-181
- [8] Li Bin, Ma Xiao-Chuan, Yan She-Feng, Li Yang. A sonar array processing system based on multi-core DSPs//Proceedings of the IEEE 11th International Conference on Signal Processing. Beijing, China, 2012: 421-424
- [9] Zhang Feng, Wu Qin-Zhang, Ren Guo-Qiang. A real-time capture system for high-resolution measure image//Proceedings of the 3rd International Workshop on Advanced Computational Intelligence. Suzhou, China, 2010: 653-656
- [10] Xie Shui-Ying. The video image processing system based on FPGA//Proceedings of the International Conference on Circuit and Signal Processing. Shenzhen, China, 2010: 727-730
- [11] Lei Guang. Design and Verification of Video Processing Module based on FPGA and Dual DSP Structure [M. S. dissertation]. School of Automation Engineering, Chengdu, 2010(in Chinese)
(雷光. 基于 FPGA 和双 DSP 结构的视频处理模块设计及验证[硕士学位论文]. 电子科技大学, 成都, 2010)
- [12] Zhang Kai, Chen Shu-Ming, Wang Yao-Hua, Ning Xi. Design tradeoffs of high performance DSPs for general-purpose HPC. Chinese Journal of Computers, 2013, 36(4): 790-798(in Chinese)
(张凯, 陈书明, 王耀华, 宁希. 面向通用 HPC 的高性能 DSP 设计权衡. 计算机学报, 2013, 36(4): 790-798)
- [13] Xia Fei, Dou Yong, Song Jian, Lei Guo-Qing. Fine-Grained Parallel RNA Secondary Structure Prediction Using SCFGs on FPGA. Chinese Journal of Computers, 2010, 33(5): 797-812(in Chinese)
(夏飞, 窦勇, 宋健, 雷国庆. 基于 FPGA 的细粒度并行 CYK 算法加速器设计与实现. 计算机学报, 2010, 33(5): 797-812)
- [14] Li Le, Xiong Zhi-Hui, Wang Bin, Zhang Mao-Jun. High-speed data communication between DSP and FPGA in embedded panoramic video processing system. Journal of Electronics & Information Technology, 2010, 32(3): 649-654(in Chinese)
(李乐, 熊志辉, 王斌, 张茂军. DSP+FPGA 折反射全景视频处理系统中双核高速数据通信. 电子与信息学报, 2010, 32(3): 649-654)
- [15] Li Bo, Ge Bao-Shan, Li Wei, Yao Chun-Lian. A general DSP based multi-modal video encoder. Chinese Journal of Computers, 2004, 27(12): 1648-1656(in Chinese)
(李波, 葛宝珊, 李炜, 姚春莲. 基于通用 DSP 的多模式视频编码器. 计算机学报, 2004, 27(12): 1648-1656)
- [16] Gao Yang, Liu Rong-Ke, Hu Wei. Design and implementation of high definition video image system based on FPGA+DSP. Electronic Measurement Technology, 2011, 34(1): 69-73(in Chinese)
(高杨, 刘荣科, 胡伟. 基于 FPGA+DSP 的高清视频图像系统设计及实现. 电子测量技术, 2011, 34(1): 69-73)
- [17] Duan Jing-Hong, Deng Ya-Ling, Liang Kun. Development of image processing system based on DSP and FPGA//Proceedings of the 8th International Conference on Electronic Measurement and Instruments. Xi'an, China, 2007: 791-794
- [18] Shen Yan-Fei, Li Jin-Tao, Zhu Zhen-Min, Zhang Yong-Dong. High efficiency video coding. Chinese Journal of Computers, 2013, 36(11): 2340-2355(in Chinese)
(沈燕飞, 李锦涛, 朱珍民, 张勇东. 高效视频编码. 计算机学报, 2013, 36(11): 2340-2355)
- [19] Jiang Hong-Xu, Zhang Xiao-Hong, Zhao Wei. Study on signal integrity simulation for high speed serial RapidIO//Proceedings of the 2nd International Conference on Green Communications and Networks (GCN). Chongqing, China, 2012: 881-886



JIANG Hong-Xu, born in 1976, Ph.D., associate professor. His research interests include digital media processing and embedded system design.

LIU Ting-Shan, born in 1986, Ph.D. candidate. Her research interests include digital media processing and embedded system design.

Background

In recent years, with the development of digital video processing application, video processing algorithms are becoming more and more complex. The complexity of video processing algorithms put forward higher request to processors. However, many high performance DSPs do not have video ports, so FPGA + DSP hybrid processor structures are widely used in video processing systems. In this processor structure, FPGA is used as video port, so we can choose higher performance DSPs.

Multi-channel and high-definition video have been the focus in video processing field. The development of digital video processing applications has made large amounts of data a new challenge to video processing systems. And it is difficult for the external memory interface (EMIF), which is widely used in most video processing systems, to transmit such large amounts of data. High speed serial interface SRIO as an important branch of RapidIO, providing the interconnection between heterogeneous processors another possibility. Especially, the integration of SRIO interface in many embedded processors, such as DSP and FPGA, to promote the development of the SRIO application. Although SRIO has several transmission rates such as 1.25 Gbps, 2.5 Gbps, 3.125 Gbps,

LI Hui-Yong, born in 1983, Ph.D. candidate. His research interests include digital media processing and embedded system design.

ZHANG Ping, born in 1987, M.S. candidate. Her research interests include digital media processing and embedded system design.

DUAN Mi-Yi, born in 1953, Ph.D., professor. His research interests include digital video/image compression processing technology, information fusion technology etc.

etc, but these are theoretical value which can be affected by inefficient transmission.

This work aimed at video data format, FPGA + DSP video processing system. We proposed a high-efficiency data transmission method between FPGA and DSP processors based on SRIO interface. In this paper we made FPGA as a core unit of transmission and DSP as for processing. Specially, we reorganized the video transmission sequence after separating the three components of video data and applied a lightweight SRIO streaming write transaction (SWRITE) which has minimal header overhead to increase data throughput. To reduce the DSP transmission time, the video data were directly stored in the predefined DSP storage unit and a simple DOORBELL transaction was used as a notification message. The experimental results show that our method can save lots of FPGA resources and at the same time the transmission rate can achieve 81% of the SRIO theoretical value, which overcomes most of existing methods. Our work is supported by the National Natural Science Foundation of China (No.61272347), which aims to improve the performance of video processing system and the National Science Fund for Distinguished Young Scholars (No. 61125206).